Communication and Network, 2010, 2, 54-61
doi:10.4236/cn.2010.21008 Published Online February 2010 (http://www.scirp.org/journal/cn)

Scientific Research

Designing Intrusion Detection System for Web Documents Using Neural Network

Hari Om, Tapas K. Sarkar
Department of Computer Science and Engineering, Indian School of Mines, Dhanbad, India
E-mail: *hariom63@rediffmail.com, aastitva@gmail.com*
Received November 17, 2009; *accepted December* 29, 2009

Abstract: Cryptographic systems are the most widely used techniques for information security. These systems however have their own pitfalls as they rely on prevention as their sole means of defense. That is why most of the organizations are attracted to the intrusion detection systems. The intrusion detection systems can be broadly categorized into two types, Anomaly and Misuse Detection systems. An anomaly-based system detects computer intrusions and misuse by monitoring system activity and classifying it as either normal or anomalous. Misuse detection systems can detect almost all known attack patterns; they however are hardly of any use to detect yet unknown attacks. In this paper, we use Neural Networks for detecting intrusive web documents available on Internet. For this purpose Back Propagation Neural (BPN) Network architecture is applied that is one of the most popular network architectures for supervised learning. Analysis is carried out on Internet Security and Acceleration (ISA) server 2000 log for finding out the web documents that should not be accessed by the unauthorized persons in an organization. There are lots of web documents available online on Internet that may be harmful for an organization. Most of these documents are blocked for use, but still users of the organization try to access these documents and may cause problem in the organization network.

Keywords: intrusion detection system, neural network, back propagation network, anomaly detection, misuse detection

1. Introduction

The information is the most important resource that must be managed efficiently. Besides management, its protection is also very important as it may lead to economic losses in today's electronic environment. For example, we can control our bank accounts from almost anywhere in the world using a suitable network, such as satellite and cellular phone networks to interact with the bank representatives, or the specialized wired ATM networks and the Internet for online banking services. The services supported by networks are very much useful and efficient, but these can be subverted by unscrupulous elements for their own benefits. So, suitable mechanism needs be employed to protect the information. In a survey of fraud against auto teller machines [1], it is reported that the patterns of fraud depends on those who were responsible for implementing and managing the systems. In USA, if a customer disputes a transaction, this is the responsibility of the bank to prove that the customer is mistaken or lying. This forced the US banks to protect their systems properly. But, in Britain, Norway and the Netherlands, the burden of proof lies on the customer. The bank is right if the customer could not prove it wrong. That is why the banks in these countries became careless. Eventually, epidemics of fraud demolished their satisfaction and in the meanwhile the US banks suffered much less fraud. Though they spent less money on security than their European counterparts, yet they spent it more effectively [2]. A different kind of incentive failure was also seen in early 2000 with distributed denial of service attacks against a number of high profile websites. Those attacks exploited a number of weak machines to launch a large coordinated packet flood at a host. Since many of them flooded the victim at the same time, the traffic was more than the host could handle. Furthermore, because it came from many different sources, it could be very difficult to stop. Varian [3] discusses different kind attacks and their effects. The suggestions made in [3] are: the costs of distributed denial-of-service attacks should fall on the operators of the networks from which the flooding traffic originates. And assign legal liability to the parties that are best able to manage the risk as they will develop expertise for computer security and provide the required services to their clients. In next section we review the intrusion detection systems.

2. Early Intrusion Detection System

An intrusion occurs when an attacker gains unauthorized access to a valid user's account and performs disruptive behavior while masquerading as that user. The attacker may harm the user's account directly or can use it to launch attacks on other accounts or machines. In such scenario a useful method to detect it is to develop "patterns" of users of a computer system. The early intrusion detection efforts used to do manual review of a system audit trail that was inefficient approach as many systems did not collect enough data to provide an audit trail, or failed to protect the data against modification. Studies in [4] show that nearly all large corporations and most medium-sized organizations have installed some form of intrusion detection tool. In [5], the misuse detection methods using mobile agents are discussed. The methods to detecting intrusions can be anomaly detection or misuse detection. Misuse detection is mainly suitable for reliably detecting known patterns, but they are hardly of any use yet unknown attack methods. The mobile agents provide computational security by constantly moving around the Internet and propagating rules to solve misuse detection. The paper [6] discusses an Intrusion Detection System (IDS) architecture integrating both anomaly and misuse detection approaches. This architecture consists of three main modules: an anomaly detection module, a misuse detection module, and a decision support system module. The anomaly detection module uses a Self-Organizing Map (SOM) structure to model normal behavior and any deviation from the normal behavior is considered as an attack. The misuse detection module uses J.48 decision tree algorithm to classify different types of attacks. The decision support system analyzes and interprets the results for interpreting the results of both anomaly and misuse detection modules. In [7], strict anomaly detection method is discussed that uses the neural networks to a great effect. Now we review the important approaches used in the intrusion detection systems.

2.1 Rule Based Intrusion Detection Systems

The basic assumption in the rule-based intrusion detection systems is that the intrusion attempts can be characterized by sequences of user activities that lead to compromised system states and based on that they predict intrusion. These systems fire rules when audit records or system status information begins to indicate illegal activity. Two major approaches are followed in rule-based intrusion detection: state-based and model-based approach. In the former, the rule base is codified using the terminology found in the audit trails and Intrusion attempts are the sequences of system state as defined by audit trail information leading from an initial and limited access state to a final compromised state [8]. In the later, the known intrusion attempts are modeled as sequences of user behavior. The intrusion detection system itself is responsible for determining how an identified user behavior may manifest itself in an audit trail. These systems have many benefits, such as large data processing, more intuitive explanations of intrusion attempts, and prediction of future actions. The rule-based systems however have some limitations. They lack flexibility in the rule-to-audit record representation. Slight variations in an attack sequence can affect the activity-rule comparison up to that extent that the intrusion may not be detected. While increasing the level of abstraction of the rule-base does provide a partial solution to this weakness, it also reduces the granularity of the intrusion detection device. A number of non-expert system-based approaches to intrusion detection have been discussed in [9–12]. Most current approaches to detecting intrusions utilize some form of rule-based analysis. Expert systems are the most common form of rule-based intrusion detection approaches [13–16]. An Expert system consists of a set of rules that encode the knowledge of a human "expert". These rules are used by the system to make conclusions about the security-related data from the intrusion detection system. Unfortunately, the expert systems require frequent updates to remain current. While the expert systems offer an enhanced ability to review audit data, the required updates may be ignored or performed infrequently by the administrator. At a minimum, this leads to an expert system with reduced capabilities. At worst, this will degrade the security of the entire system by causing the system's users to be mislead into believing that the system is secure, even as one of the key components becomes increasingly ineffective over the time.

2.2 Network-Based and Host-Based Intrusion Detection Systems

A network-based intrusion detection system (NIDS) observes the traffic at specified points in the network and then checks that traffic packet by packet in real time to detect intrusion patterns. It can examine the activity at any layer of the network such as network layer, transport layer, and application layer protocol. The network-based systems are generally best at detecting the unauthorized outsider access and bandwidth theft/denial of service. When an unauthorized user logs in successfully, or attempts to log in, they are tracked with host-based IDS. However, detecting the unauthorized users before their logon attempt is best accomplished with network-based IDS. The packets that initiate bandwidth theft attacks can best be noticed with use of network-based IDS. Some of the network-based IDS are Shadow, Dragon, NFR, RealSecure, and NetProwler.

Host-based Intrusion Detection systems are first of IDSs developed and implemented. They collect and analyze the data originated on a computer that provides a

service, such as web server. After collecting the data from a given computer, it is analyzed. One example of the host-based system is programs that operate on a system and receive application or operating system audit logs. These programs are highly effective for detecting insider abuses. Residing on the trusted network systems themselves, they are close to the network's authenticated users. If one of these users attempts an unauthorized activity, the host-based systems usually detect and collect the most pertinent information in the quickest possible manner. In addition to detecting unauthorized insider activity, the host-based systems are also effective at detecting unauthorized file modification. The host-based IDSs are Windows NT/2000 Security Event Logs, RDMS audit sources, Enterprise Management systems audit data (such as Tivoli), and UNIX Syslog in their raw forms.

Graph-Based Intrusion Detection System (GrIDS) [17] uses a graphical representation to monitor the activity of entire network. EMERALD eXpert-BSM, a real-time forward-reasoning expert system, uses a knowledgebase to detect multiple forms of system misuse [18]. In [19], a technique is discussed for detecting intrusions at the level of privileged processes. It is reported that short sequences of system calls executed by running programs are a good discriminator between normal and abnormal operating characteristics of several common UNIX programs. Analyzing the system calls made by a program is a reasonable approach to detect intrusions based on program behavior profiles [20].

2.3 Neural Network Based Intrusion Detection Systems

The neural network based intrusion detection systems have the ability to be trained and learn patterns in a given environment, which can be used to detect intrusions by recognizing patterns of an intrusion. The Artificial Neural Network based methods for intrusion detection are quite popular. Recently an investigation on the unsupervised neural network models and choice for most appropriate one among them for evaluation and implementation is discussed in [21]. These can be used for both host-based and network based intrusion detection systems. For the success of IDS is the failure of firewalls to prevent many security intrusions. The intrusion detection systems can detect many of them that slip through firewalls. Many Anomalies based and Misuse based intrusion detection techniques have been designed to detect the abnormal behavior exhibited by the user in [22–27]. Artificial neural networks have been suggested as alternatives to the statistical analysis [28–30]. Statistical Analysis involves statistical comparison of current events to a predetermined set of baseline criteria. Neural networks are specifically discussed to identify the typical characteristics of system users and identify statistically significant variations from the user's established behavior. Artificial neural networks have also been discussed for use in the detection of computer viruses. In [31], neural networks are discussed as statistical analysis approaches in the detection of viruses and malicious software in computer networks. The neural network intrusion detection (NNID) system [32] uses neural networks to predict the next command a user will enter based on previous commands. Now we discuss our neural network based intrusion detection system.

3. Audit Logs Analysis Using Neural Networks

In this work, we collect the data from the ISA 2000 Web Access Log to analyze for possible intrusion attacks using the neural networks and then use the back propagation neural (BPN) network model for analyzing the input data. Different numbers of hidden layers are considered in the PBN algorithm.

3.1 ISA 2000 Web Access Log Analysis

Internet bandwidth is consumed by a variety of internet application protocols. The most popular application layer protocol that accesses Internet resources is the HTTP protocol. It is used to access the resources on the World Wide Web. Although bandwidth cost per-kilobyte or per-megabyte has come down over the years, yet the amount of bandwidth consumed by users on the campus network increases year after year. HTTP connections to Internet resources not only lead to increase in bandwidth usage, they also reduce the amount of bandwidth available on the Internet link for other important protocols and applications, such as SMTP, POP3 and VPN. In order to provide the desired data resources to users, it is stored at different locations using some kind of servers. To further help the user in computer network environment, proxy servers are employed. A proxy server is a server (a computer system or an application program) which provides the services to user requests by making requests to other servers. A user connects to the proxy server, requesting a file, connection, web page, or other resource available from a different server. In an enterprise that uses the Internet, a proxy server is a server that acts as an intermediary between a workstation user and the Internet so that the enterprise can ensure security, administrative control, and caching service. It can receive a request for an Internet service (such as a Web page request) from a user. On clearing filtering requirements, the proxy server, assuming it is also a cache-server, looks in its local cache of previously downloaded Web pages. If the desired pages are there, it returns them to the user without needing to forward the request to the Internet. In case the required pages are not in the cache, the proxy server, acting as a client on behalf of the user, uses one of its own IP addresses to request the pages

Table 1. Attributes in ISA server 2000 log file

Field name	Description
c-ip	The *Internet Protocol (IP)* address of the requesting client.
cs-username	The account of the user making the request. If ISA Server access control is not being used, ISA Server uses Anonymous.
c-agent	The name and version of the client application sent by the client in the *Hypertext Transfer Protocol (HTTP)* User-Agent header.
date	The date on which the logged event occurred.
time	The local time when the logged event occurred.
r-host	The domain name for the remote computer that provides service to the current connection.
r-ip	The network IP address of the remote computer that provides service to the current connection.
r-port	The reserved *port number* on the remote computer that provides service to the current connection.
time-taken	The total time, in milliseconds, that is needed by ISA Server to process the current connection
cs-bytes	The number of bytes sent from the remote computer and received by the client during the current connection.
sc-bytes	The number of bytes sent from the client to the remote computer during the current connection.
cs-protocol	The application protocol used for the connection. Common values are http for Hypertext Transfer Protocol, https for Secure HTTP, and ftp for *File Transfer Protocol*.
s-operation	The HTTP method used. Common values are GET, PUT, POST, and HEAD.
cs-uri	The URL requested.
s-object-source	The type of source that was used to retrieve the current object. A table of some possible values is provided in Object Source Values.
sc-status	A Windows (Win32) error code (for values less than 100), an HTTP status code (for values between 100 and 1,000), a Winsock error code (for values between 10,004 and 11,031), or an ISA Server error code.

from the server out on the Internet. When the pages are received, the proxy server forwards them onto the user.

3.2 ISA Server 2000 Web Access Log

Internet Security and Acceleration (ISA) Server 2000 can help in reducing overall bandwidth usage and cost by caching Web contents on the ISA Server 2000. We use Microsoft ISA Server 2000 log to monitor and analyze the status of the Web proxy requests to find out the documents that are worthless in an organization. Table 1 shows the attributes used in ISA Server 2000 Log file.

3.3 Experiment

The input data is collected in terms of above mentioned attributes. Table 2 contains the values of the input data.

The data shown in Table 2 is not a valid input pattern for BPN. Before providing the data for training to the BPN, it needs be converted in the valid pattern. We perform the following steps for making a valid input for BPN.

- Select the ip address part of the destination web server and convert it in the integer number without delimiter. For example, the ip 216.239.63.83 is converted into 2162396383. This is a long number which in itself is not a valid input pattern for BPN.
- Normalize the input pattern in real numbers. After normalization the input data pattern is shown in Table 3. First column shows the normalized ip addresses and the second column shows 1 as valid ip address and 0 as invalid ip addresses.
- Train the BPN for this input pattern by taking dif ferentnumber of hidden layers. We use 2, 5 and 10 hidden layers. The number of epochs is taken as 50,000. Results

Table 2. ISA server 2000 web access log

c-ip	cs-user-name	c-agent	date	time	r-host	r-ip	r-port	Time-ta	cs-bytes	sc-bytes	cs-protocol	s-op-eration	cs-uri	s-objece-source	sc-stauts
10.0.4.36	anonymous	Mozilla/4	12/14/2006	7:01.41	Images3.0	72.14.209	80	797	796	3053	http	GET	http://image	VCache	30
10.0.4.46	Anonymous	Mozilla/5	12/14/2006	7:01.41	www.orku	72.14.209	80	797	981	253	http	GET	http://www	Inet	30
10.0.4.46	Anonymous	Mozilla/5	12/14/2006	7:01.41	Images3.0	72.14.209	80	813	995	253	http	GET	http://image	VCache	30
10.0.14.23	Anonymous	Mozilla/4	12/14/2006	7:01.41	Immail.re	210.161.32	80	640	1243	237	http	GET	http://image	Inet	30
10.0.7.221	Anonymous	Mozilla/4	12/14/2006	7:01.41	In.f89.mail	203.84.222	80	5844	2332	79644	http	POST	http://in.f89	Inet	20
10.0.4.123	Anonymous	Mozilla/5	12/14/2006	7:01.41	www.orku	72.14.209	80	3109	1135	7267	http	GET	http://www	Inet	20
10.0.4.165	Anonymous	Mozilla/4	12/14/2006	7:01.41	Jdelivery	210.161.32	80	593	1014	277	http	GET	http:// jesliv	Inet	30
10.0.98.43	Anonymous	Mozilla/4	12/14/2006	7:01.41	In.wrs.yal	216.252.12	80	1359	816	601	http	GET	http://in,wrn	Inet	30
10.0.4.185	Anonymous	Mozilla/4	12/14/2006	7:01.41	Mum.inte	220.226.20	80	4531	358	2312	http	GET	http:// mum	Inet	00
10.0.4.46	Anonymous	Mozilla/5	12/14/2006	7:01.41	Imagas3.0	72.14.209	80	797	995	253	http	GET	http://image	VCache	30
10.0.4.46	Anonymous	Mozilla/5	12/14/2006	7:01.41	Imagas3.0	72.14.209	80	781	1000	253	http	GET	http://image	VCache	30
10.0.4.36	Anonymous	Mozilla/4	12/14/2006	7:01.41	Imagas3.0	72.14.209	80	766	796	2257	http	GET	http://image	VCache	30
10.0.4.36	Anonymous	Mozilla/4	12/14/2006	7:01.42	Imagas3.0	72.14.209	80	781	796	2215	http	GET	http://image	VCache	30
10.0.4.179	Anonymous	Mozilla/4	12/14/2006	7:01.42	www.goo	72.14.235	80	859	969	1532	http	GET	http://www	Inet	20
10.0.4.163	Anonymous	Mozilla/4	12/14/2006	7:01.42	Images3.0	72.14.209	80	1563	747	1882	http	GET	http://image	Inet	20
10.0.4.36	Anonymous	Mozilla/4	12/14/2006	7:01.42	www.orku	72.14.209	80	5312	968	18229	http	GET	http://www	Inet	20
10.0.4.36	Anonymous	Mozilla/4	12/14/2006	7:01.42	Images3.0	72.14.209	80	797	994	2281	http	GET	http://image	VCache	30
10.0.4.54	Anonymous	Mozilla/4	12/14/2006	7:01.42	www.orku	72.14.209	80	3953	1013	18507	http	GET	http://www	Inet	20
10.0.4.46	Anonymous	Mozilla/5	12/14/2006	7:01.42	Images3.0	72.14.209	80	796	1014	201	http	GET	http://image	VCache	30
10.0.4.46	Anonymous	Mozilla/5	12/14/2006	7:01.42	Images3.0	72.14.209	80	812	1008	201	http	GET	http://image	VCache	30
10.0.4.165	Anonymous	Mozilla/4	12/14/2006	7:01.42	jdelivery	210.161.32	80	594	1030	276	http	GET	http://jdeliv	VCache	30
10.0.4.39	Anonymous	Mozilla/4	12/14/2006	7:01.42	www2.nu	69.25.142	80	133125	1683	1119	http	GOST	http://www	Inet	6
10.0.4.174	Anonymous	Mozilla/4	12/14/2006	7:01.42	Mail.goog	209.85.139	80	2563	1683	361	http	GET	http://mail	Inet	20
10.0.4.193	Anonymous	Mozilla/4	12/14/2006	7:01.42	www.go	72.14.235	80	703	340	234	http	GET	http://www	VCache	30
10.0.4.46	Anonymous	Mozilla/5	12/14/2006	7:01.42	Images3.0	72.14.209	80	781	1013	201	http	GET	http://image	VCache	30
10.0.4.46	Anonymous	Mozilla/5	12/14/2006	7:01.42	Images3.0	72.14.209	80	797	1014	201	http	GET	http://image	VCache	30
10.0.4.36	Anonymous	Mozilla/4	12/14/2006	7:01.42	Images3.0	72.14.209	80	766	796	2330	http	GET	http://image	VCache	30

Table 3. Normalized training patterns for BPN

Normalized IP addresses	Valid(0) / Invalid(1)	Normalized IP addresses	Valid(0) / Invalid(1)
0.549298	0	0.815306	0
0.57671	0	0.819334	0
0.588196	0	0.819424	0
0.753141	0	0.819514	0
0.760483	0	0.819537	0
0.780321	0	0.026241	1
0.780564	0	0.007997	1
0.791906	0	0.027761	1
0.795886	0	0.28298	1
0.803925	0	0.002624	1
0.803937	0	0.0819573	1
0.808023	0	0.000331	1
0.811643	0	0.081742	1
0.81187	0	0.076052	1
0.811933	0		

for different number of hidden layers are shown in Table 5.

- After training the BPN, it is tested with test patterns as shown in Table 4.

4. Results

The training of the neural networks has been conducted using the Back Propagation neural network algorithm for 50,000 iterations of the selected training data. After training the BPN, the following results are obtained.

The results obtained match very closely with the desired root mean square (RMS) error as shown in Table 5. Though this method is not designed to be used as a complete intrusion detection system, yet the results show the potential of neural networks to detect individual instances of possible misuse from a representative web-based data. Graphs in Figure 1 show the results for different number of hidden layers used in the BPN. It is evident from the graphs that the results are very close to desired output values, when we use 10 numbers of neurons for hidden layer.

5. Discussions

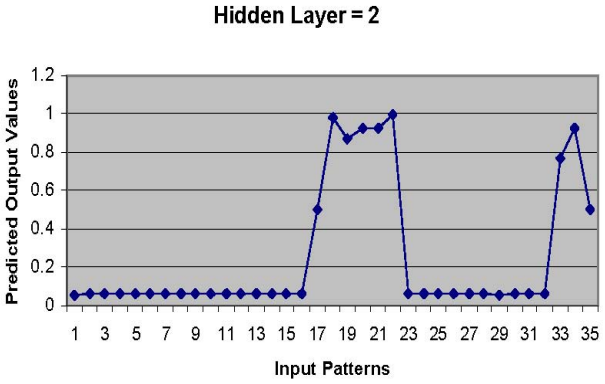
The above mentioned method can be used to find out the web documents that should not be allowed in the organization. Web Server log file is divided into two parts. One file contains only the destination ip addresses and the second file contains the corresponding source ip and date

Table 4. Normalize testing patterns for BPN

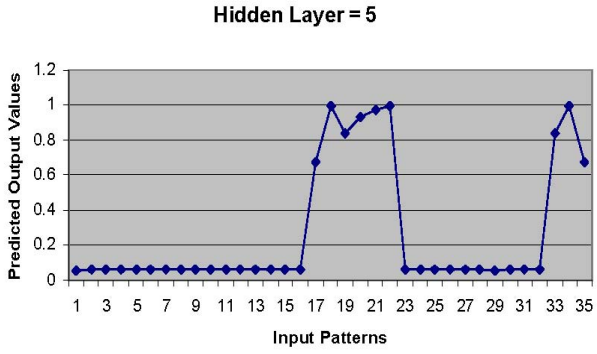
IP Patterns for Testing	Valid(0) / Invalid(1)	IP Patterns for Testing	Valid(0) / Invalid(1)
0.000771	0	0.00082	0
0.000776	0	0.000823	0
0.000788	0	0.000826	0
0.000793	0	0.000831	0
0.000794	0	0.819573	1
0.000795	0	0.000331	1
0.000796	0	0.081742	1
0.000798	0	0.002624	1
0.000799	0	0.076052	1
0.0008	0	0.259212	1
0.000802	0	0.008221	1
0.000813	0	0.027213	1
0.000818	0	0.000819	1

Table 5. RMS error corresponding to hidden layers

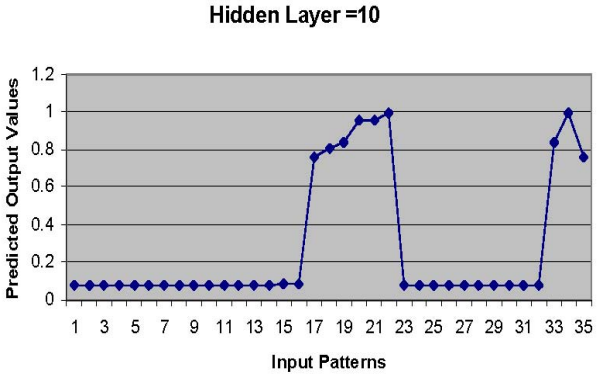
No of Hidden Layers	RMS Error (Training Data)
2	0.026315
5	0.024311
10	0.023302



(a)



(b)



(c)

Figure 1. Predicted output for test patterns: in (a) 2, in (b) 5, and in (c) 10 hidden layers are used

and time of the site being accessed. Input of the first file having ip addresses of the sites being accessed is converted into normalized ip address. This is the input pattern to Neural Network for testing. For the ip addresses having errors (invalid websites) and no errors (valid websites) the Neural Network is already trained. When a user tries to access a website that is in the invalid website record, it is detected by the system. At the time there is a

Table 6. Web site address to be included in the invalid web site record

Address of the Web Site	ip address
www.bollyexpress.com	208.101.17.60
www.maxalbums.com	64.246.28.216

deviation in the log files under testing it will be figured out. Here in our case Normalized ip pattern 0.002624 is reported as invalid and its corresponding website is www.mp3fine.com. The corresponding source ip address, time, and date can be found from the second file.

We have manually analyzed Web Server log for duration of 15 minutes after the first detection is reported in the system. This is because there is a probability that the user on the system may try to access some similar sites that should be in the invalid web site record, but are not included in the invalid website record previously. This analysis gives us positive results and two sites have been included in the invalid website record as mentioned in Table 6.

There are lots of web documents which provide anonymous downloads of the files of larger size like movie and songs files. If a user is allowed to access these sites, then a large portion of the network bandwidth will be wasted. Many of the sites are already blocked by the Network Administrator, but some sites are still in use. When a user is stopped to access a web document he/she will try to access another web document with similar facility that is missed to block by the Network Administrator. The analysis discussed above can be used to block these types of Web documents.

6. Conclusions

Research and development of intrusion detection system has been ongoing last couple of decades and the challenges faced by designers have increased many fold. Misuse detection is particularly difficult problem because of the extensive number of vulnerabilities in computer systems and the creativity of the attackers. Neural networks provide a number of advantages in the direction of these attacks. The results of our tests for the Proxy Server (Microsoft ISA Server 2000) log show that this technique can be applied for detecting worthless web document access to save the network bandwidth.

REFERENCES

[1] R. J. Anderson, “Why cryptosystems fail,” In Communications of the ACM, Vol. 37, No. 11, pp. 32–40, November 1994.

[2] http://www.cert.org/reports/dsit_ workshop-final.html.

[3] H. Varian, “Managing online security risks,” Economic Science Column, The New York Times, June 2000.

[4] SANS Institute staff, “Intrusion detection and vulnerability testing tools: what works?” 101 Security Solutions E-Alert Newsletters, 2001.

[5] T. K. Kim, D. Y. Lee, and T. M. Chung, “Mobile agent-based misuse intrusion detection rule propagation model for distributed system,” Lecture Note in Computer Science, Vol. 2510, pp. 842–849, 2002.

[6] O. Depren, M. Topallar, E. Anarim, and M. K. Ciliz, “An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks,” Expert Systems with Applications, Vol. 29, No. 4, pp. 713–722, November 2005.

[7] T. Konno and M. Tateoka, “Accuracy improvement of anomaly-based intrusion detection system using taguchi method,” Proceeding of Symposium on Applications and the Internet Workshops (SAINT-W’05), 0-7695-2263-7/05, 2005.

[8] K. Ilgun, “USTAT: A real-time intrusion detection system for UNIX,” Proceeding of the 1993 Computer Society Symposium on Research in Security and Privacy, Oakland, California, Los Alamitos, pp. 16–28, May 1993.

[9] K. Fox, R. Henning, J. Reed, and R. Simonian, “A neural network approach towards intrusion detection,” Proceeding of 13th National Computer Security Conference, Baltimore, MD, pp. 125–134, 1990.

[10] J. Frank, “Artificial intelligence and intrusion detection: current and future directions,” Computers and Security, Vol. 14, No. 1, pp. 31–31(1), 1995.

[11] L. Fu, “A neural network model for learning rule-based systems,” Proceeding of the International Joint Conference on Neural Networks, pp. 343–348, 1992.

[12] D. Hammerstrom, “Neural networks at work,” IEEE Spectrum, pp. 26–53, June 1993.

[13] J. Zimmermann, L. Mé, and C. Bidan, “An improved reference flow control model for policy-based intrusion detection,” Proceeding of the 8th European Symposium on Research in Computer Security (ESORICS), pp. 291–308, October 2003.

[14] G. J. Nalepa, “Application of the XTT rule-based model for formal design and verification of internet security systems,” Lecture Notes in Computer Science, Vol. 4680, pp. 81–86, 2007.

[15] D. Dorothy, “An intrusion-detection model,” IEEE Transactions on Software Engineering, Vol. 13, No. 2, pp. 222–232, February 1987.

[16] M. M. Sebring, E. Shellhouse, M. E. Hanna, and R. A. Whitehurst, “Expert systems in intrusion detection: a case study,” Proceeding of the 11th National Computer Security Conference, Baltimore, MD, pp. 74–81, October 1988.

[17] S. Staniford-Chen, S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, C. Wee, R. Yip, and D. Zerkle. “GrIDS, a graph based intrusion detection system for large networks,” Proceeding of the 20th National Information Systems Security Conference, Vol. 1, pp. 361–370, October 1996.

[18] P. A. Porras and P. G. Neumann, “Emerald: event moni-

toring enabling responses to anomalous live disturbances," Proceeding of the 20th National Information systems Security Conference, pp. 35–365, October 1997.

[19] S. Freeman, "Host based intrusion detection using user signatures," Computer Science Master's project, May 2002.

[20] A. K. Ghosh, A. Schwartzbard, and M. Schatz, "Learning program behavior profiles for intrusion detection," Proceeding of the 1st Workshop on Intrusion Detection and Network Monitoring, pp. 51–62, April 1999.

[21] A. ¨Oks¨uz, "Unsupervised intrusion detection system," Master Thesis, Technical University of Denmark, 2007.

[22] A. Boukerche, K. R. Lemos Juc, J. B. Sobral, and M. Sechi Moretti Annoni Notare, "An artificial immune based intrusion detection model for computer and telecommunication systems," Parallel Computing, Vol. 30, No. 5–6, pp. 629–646, 2004.

[23] R. Beghdad, "Modelling and solving the intrusion detection problem in computer networks," Computers and Security, Vol. 23, No. 8, pp. 687–696, 2004.

[24] T. F. Lunt and R. Jagannathan, "A prototype real-time intrusion-detection system," Proceeding of the Symposium on Security and Privacy, New York, pp. 59–66, April 1988.

[25] T. D. Garvey and T. F. Lunt, "Model based intrusion detection," Proceeding of the 14th National Computer Security Conference, pp. 372–385, October 1991.

[26] K. Ilgun, "Ustat: A real-time intrusion detection system for UNIX," Master's thesis, Computer Science Dept, UCSB, July 1992.

[27] S. Kumar and E. H. Spafford, "A pattern matching model for misuse intrusion detection," The COAST Project, Purdue University, 1996.

[28] J. Ryan, M. Lin, and R. Miikkulainen, "Intrusion Detection with Neural Networks," AI Approaches to Fraud Detection and Risk Management: Papers from the 1997 AAAI Workshop (Providence, Rhode Island), pp. 72–79, 1997.

[29] H. Debar and B. Dorizzi, "An application of a recurrent network to an intrusion detection system," Proceeding of the International Joint Conference on Neural Networks, pp. 478–483, 1992.

[30] A. Abraham, C. Grosan, and C. Martin-Vide, "Evolutionary design of intrusion detection programs," International Journal of Network Security, Vol. 4, No. 3, pp. 328–339, March 2007.

[31] M. Denault, D. Gritzalis, D. Karagiannis, and P. Spirakis, "Intrusion detection: approach and performance issues of the securenet system," Computers and Security, Vol. 13, No. 6, pp. 495–500, 1994.

[32] S. E. Smaha, "Haystack: an intrusion detection system," Proceeding of the Fourth AeroSpace Computer Security Applications Conference, Orlando, FL, pp. 37–44, December 1988.

